Terms used **modify executable runtime**

Found **14** of **132,857**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 14 of 14

Relevance scale

**1** A hardware mechanism for dynamic extraction and relayout of program hot spots

Matthew C. Merten, Andrew R. Trick, Erik M. Nystrom, Ronald D. Barnes, Wen-mei W. Hmu

May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture**, Volume 28 Issue 2

Full text available: pdf(320.13 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents a new mechanism for collecting and deploying runtime optimized code. The code-collecting component resides in the instruction retirement stage and lays out hot execution paths to improve instruction fetch rate as well as enable further code optimization. The code deployment component uses an extension to the Branch Target Buffer to migrate execution into the new code without modifying the original code. No significant delay is added to the total execution of the program ...

**2** Some effects of coding styles on optimizations

Daniel Burton

December 1990 **Proceedings of the conference on TRI-ADA '90**

Additional Information: full citation, abstract

This presentation examines the effect of a few simple Ada coding styles on compiler optimizations. The discussion should make the user aware of the impact these styles have on the ability of an Ada compiler to optimize the resulting programs. During the presentation, simple examples of each style will be presented, along with the impact of each on a compiler's ability to optimize a program. First, the use of separates adversely affects the performance of optimizations across call ...

**3** Intrusion detection: Countering code-injection attacks with instruction-set randomization

Gaurav S. Kc, Angelos D. Keromytis, Vassilis Prevelakis

October 2003 **Proceedings of the 10th ACM conference on Computer and communication security**

Full text available: pdf(146.35 KB) Additional Information: full citation, abstract, references, index terms

We describe a new, general approach for safeguarding systems against *any* type of code-injection attack. We apply Kerckhoff's principle, by creating process-specific randomized instruction sets (*e.g.,* machine instructions) of the system executing potentially vulnerable software. An attacker who does not know the key to the randomization algorithm will inject code that is invalid for that randomized processor, causing a runtime exception. To

Your search matched **14** of **1034807** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

**Refine This Search:**

You may refine your search by editing the current search expression or enteri new one in the text box.

( modify<in>ab ) <and> ( executable<in>ab ) <and> (  Search

☐ Check to search within this result set

**Results Key:**

**JNL** = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **An architectural framework for runtime optimization**

*Merten, M.C.; Trick, A.R.; Barnes, R.D.; Nystrom, E.M.; George, C.N.; Gyllenh J.C.; Hwu, W.-M.W.;*

Computers, IEEE Transactions on , Volume: 50 , Issue: 6 , June 2001
Pages:567 - 589

[Abstract] [PDF Full-Text (3812 KB)] **IEEE JNL**

---

2 **Dynamically discovering likely program invariants to support progr evolution**

*Ernst, M.D.; Cockrell, J.; Griswold, W.G.; Notkin, D.;*

Software Engineering, IEEE Transactions on , Volume: 27 , Issue: 2 , Feb. 20
Pages:99 - 123

[Abstract] [PDF Full-Text (776 KB)] **IEEE JNL**

---

3 **Environment for multiprocessor simulator development**

*Wakabayashi, M.; Amano, H.;*

Parallel Architectures, Algorithms and Networks, 2000. I-SPAN 2000. Proceed International Symposium on , 7-9 Dec. 2000
Pages:64 - 71

[Abstract] [PDF Full-Text (672 KB)] **IEEE CNF**

---

4 **A visual framework for the scripting of parallel agents**

*Mueller, W.; Meyer, A.; Zabel, H.;*

Visual Languages, 2000. Proceedings. 2000 IEEE International Symposium on 13 Sept. 2000
Pages:77 - 78